ABSOLUTE BINARY PROGRAM NO. 12943-16001 AND 12943-16002
DATE CODE 1728 AND 1432

EXTENDED INSTRUCTION GROUP DIAGNOSTIC

reference manual

For HP 1000 Series Computers

HEWLETT-PACKARD COMPANY
11000 WOLFE ROAD, CUPERTINO, CALIFORNIA, 95014

MANUAL PART NO. 12943-90004
MICROFICHE PART NO. 12943-90005

Printed: SEPT 1978
Printed in U.S.A.

NOTICE

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

Library Index Number
2DIAG.070.12943-90004

CONTENTS

ILLUSTRATIONS

TABLES

INTRODUCTION

SECTION I

1-1. GENERAL

The Extended Instruction Group Diagnostic (EIG) tests the operation of the HP 12943 Extended Instruction Set for HP 1000 M-Series, E-Series and F-Series Computers. The diagnostic is divided into two 4K programs which check: (a) index instructions, and (b) word, byte, and bit instructions. These instruction sets are indicated and discussed separately in the manual wherever necessary.

The following diagnostic should be run before running this diagnostic:

a. Memory Reference
b. Alter Skip
c. Shift Rotate
d. Memory Protect

1-2. REQUIRED HARDWARE

The required hardware consists of the following:

a. An HP 1000 M-Series, E-Series or F-Series Computer with at least 4K-words of memory.

b. Loading device for loading the diagnostic program.

c. Console device for message reporting (recommended but not required).

d. An I/O printed circuit assembly (PCA) with interrupt logic is required in the word-byte-bit test to check the ability to interrupt.

1-3. SOFTWARE REQUIREMENTS

The following software is required:

a. The Diagnostic Configurator (part number listed below) is used for equipment configuration and as a console device driver.

   Absolute binary program, part no. 24296-60001
   Reference Manual, part no. 02100-90157.

b. Extended Instruction Group Diagnostic

	ABSOLUTE BINARY PROGRAM	DIAGNOSTIC SERIAL NO.
Index	12943-16002	101011
Word-Byte-Bit	12943-16001	101112

The diagnostic serial number (DSN) is contained in memory location 126 (octal).

PROGRAM ORGANIZATION

SECTION II

2-1. ORGANIZATION

The index instruction diagnostic is divided into 12 tests and the word-byte-bit instruction diagnostic is divided into 10 tests. These tests are listed in table 3-3.

2-2. TEST CONTROL AND EXECUTION

The diagnostics output a title message "EIG (INDEX) DIAGNOSTIC" or "EIG (WORD.BYTE.BIT) DIAGNOSTIC" to the console device (if present) for operator information and then executes the test according to the options selected on the Switch Register (see table 3-1).

The diagnostic keeps count of the number of passes that have been completed and will output the pass count at the completion of each pass (if Switch Register bit 10 is clear). At the end of each pass the computer will halt (displaying 102077) and the pass count will be in the A-register (if Switch Register bit 12 is clear). Press RUN to execute another pass. The count is cleared whenever the program is restarted.

Test sections are executed one after another in each diagnostic pass. User selection or default will determine which test sections will be executed.

2-3. SELECTION OF TEST BY OPERATOR

The operator has the capability to select his own tests or sequence of tests with the help of Switch Register bit 9. The computer halts displaying 102075 and then desired tests are selected by setting corresponding bits in the A-register. Paragraph 3-4 outlines the test selection.

2-4. MESSAGE REPORTING

There are two types of messages: error and information. Error messages are used to inform the operator when the Extended Instruction Set fails to respond correctly to a given control or sequence. Information messages are used to inform the operator of the progress of the diagnostic.

Error messages can be suppressed by setting Switch Register bit 11 and error halts can be suppressed by setting Switch Register bit 14.

Information messages are suppressed by setting Switch Register bit 10.

2-5. MESSAGE REPORTING (INDEX)

If a console device is used, the printed or displayed message will be preceded by an E (error) or H (information) and a number (in octal). The octal number is also related to the halt code when a console device is not available.

Example — Error with halt

Message: E034 LDX A FAILED

Halt Code: 102034 (T-register)

Example — Information only

Message: H047 MEMORY PROTECT OPTION NOT PRESENT

Halt Code: None

If a console device is not available, errors are processed using memory halt codes and inspecting the A-register and several memory locations for the information necessary to define the error.

The halt defines the general error. The A-register bits 0-3 contain the test number and bits 4-6 define the subtest sequence. Starting at memory location 200 (octal) the following values are stored upon error halt:

200	A-register actual
1	A-register expected
2	B-register actual
3	B-register expected
4	X-register actual
5	X-register expected
6	Y-register actual
7	Y-register expected
210	Memory actual
1	Memory expected
2	Overflow error actual
3	Extend error actual

For the case of overflow-extend error (OEACT, EEACT) bit 15 corresponds to the expected value and bit 0 to the actual value.

2-6. MESSAGE REPORTING (WORD-BYTE-BIT)

If a console device is used, the printed or displayed message will be preceded by an E (error) and a number (in octal). The octal number is also related to the halt code when a console device is not available. When an error is reported, the test-case (TC) number and instruction mnemonic is listed. The operator may then look in Appendix A and find the values used for the TC that is failing.

Example

Message: E005 WRONG RESULT RETURNED IN A-REG
IN TEST-CASE 000001 OF LBT INSTR.

Halt Code: 102005 (T-register)

The test-case (TC) determines the operand value(s) used with the instruction under test. A list of the test-cases for each instruction is shown in Appendix A.

For the example given, the operator must look in Appendix A for the LBT instruction in test-case 1. The parameters for test-case 1 are as follows:

a. The B-register* contains a left byte address (DBL).

b. The contents of the byte address in the B-register are 125377 (octal).

c. After the LBT instruction is executed, the A-register should contain the octal value 252.

*The LBT instruction increments the B-register; determine the actual byte address used by decrementing the currect contents of the B-register.

2-7. DIAGNOSTIC LIMITATIONS

The word-byte-bit diagnostic does not test the wrap around characteristic of the move byte (MBT) and move word (MVW) instructions. In addition, the ability of the scan byte instruction (SFB) to terminate at the last byte in memory, if the test or termination bytes are not present, is not checked.

OPERATING PROCEDURES

SECTION III

3-1. OPERATING PROCEDURES

A flowchart of the operating procedure is provided in figure 3-1.

3-2. RUNNING THE DIAGNOSTIC

The program will execute the diagnostic according to options selected in the Switch Register. At the completion of each pass of the diagnostic, the pass count is printed on the console device for operator information. If Switch Register bit 12 was not selected (set), the computer will halt with 102077 (octal) in the T-register. At this point, the A-register contains the pass count. To run another pass, press RUN.

3-3. RESTARTING

The program can be restarted by setting the P-register to 2000 (octal). Select Switch Register options shown in table 3-2 and press RUN.

If a trap cell halt occurs (106077 octal), the user must determine the cause of the interrupt or transfer of control to the location shown in the M-register. The program may need to be reloaded to continue.

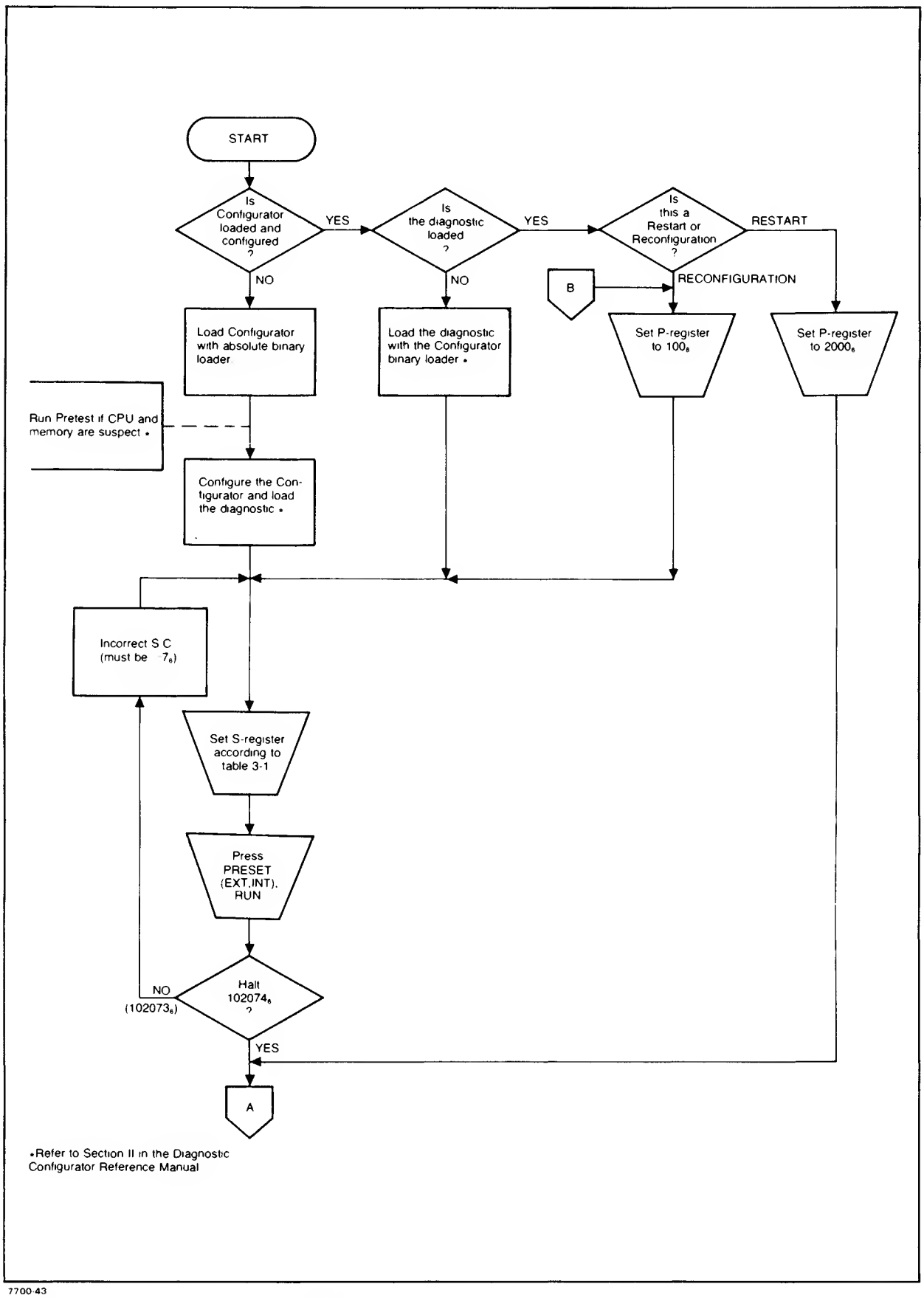
3-4. TEST SELECTION BY OPERATOR

The diagnostic provides the operator with a method to select his own test, or sequence of tests, to be run. The operator sets Switch Register bit 9 to indicate the desire to make a selection. The computer will come to a halt 102075 (octal) to indicate that it is ready for selection. If the program is running, the current test will be completed and then the program will halt. The operator then loads the A-register with the tests desired; A-register bit 0 represents Test 00, bit 1 represents Test 01, etc. (Refer to table 3-3.) The operator must then clear Switch Register bit 9 and press RUN. The operator's selection will then be run. If the operator clears all bits, the standard sequence will be run.

Table 3-1. Initial Switch Register Settings

BIT	MEANING IF SET
5-0	Select Code of any standard* I/O device used by interruptible tests.
15-6	Reserved.

* Standard I/O implies that the interface will respond to the assigned meaning of the I/O instructions and will interrupt when the Control and Flag are set and the interrupt system is enabled.



7700-43

Figure 3-1. Operating Procedure Flowchart (Sheet 1 of 2)

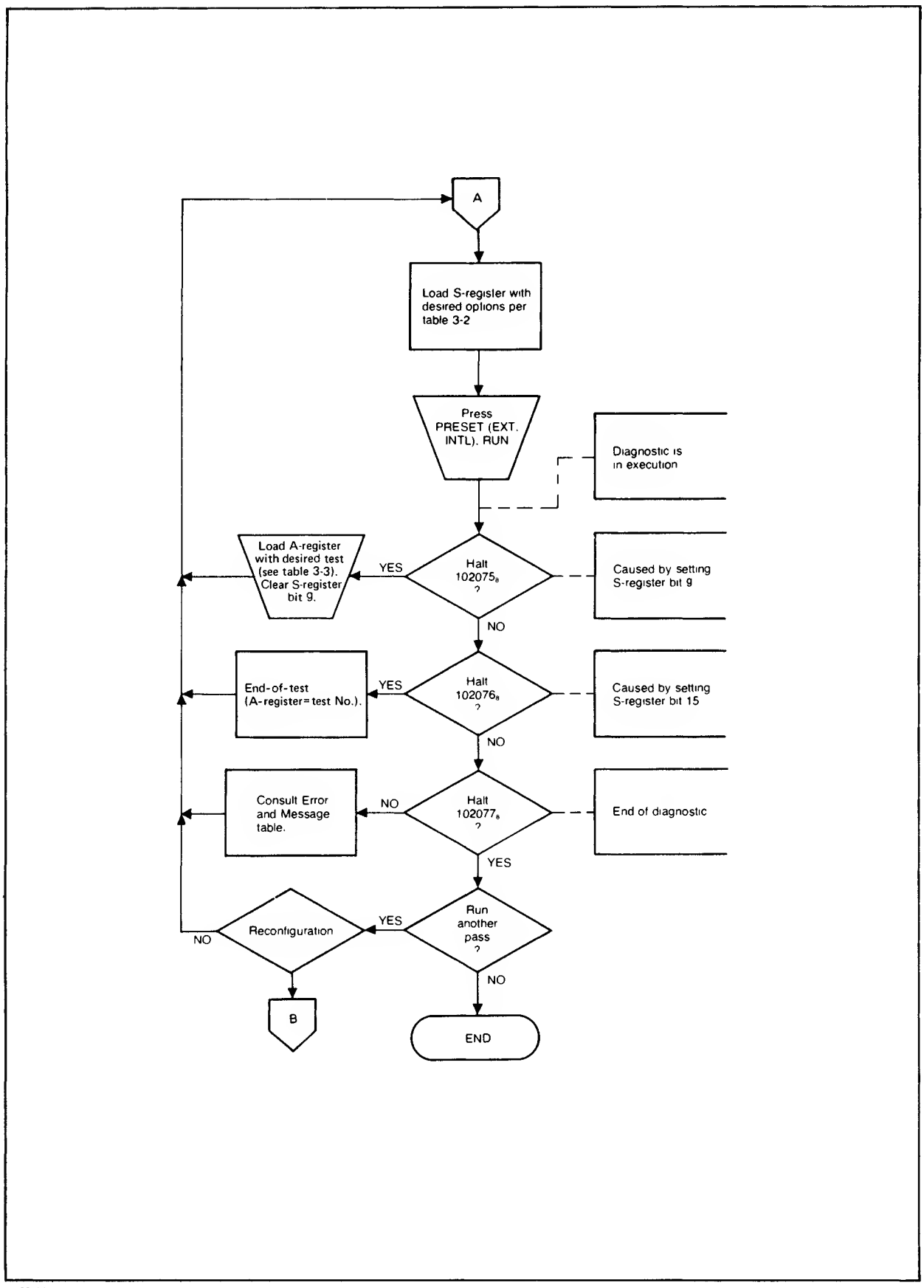


Figure 3-1. Operating Procedure Flowchart (Sheet 2 of 2)

Table 3-2. Switch Register Options

BIT	MEANING IF SET
8-0	Reserved.
9	Abort current diagnostic execution and halt (102075); user may specify a new group of tests in the A-register (see table 3-3), clear bit 9, and then press RUN.
10	Suppress non-error messages.
11	Suppress error messages.
12	Repeat all selected tests after diagnostic run is complete without halting. Message "PASS XXXXXX" will be output before looping unless bit 10 is set or console is not present. Also those tests requiring operator intervention will be suppressed.
13	Repeat last test executed (loop on test).
14	Suppress error halts.
15	Halt (102076) at the end of each test; the A-register will contain the test number in octal.

Table 3-3. Test Selection Summary

A-REG	TEST	WORD-BYTE-BIT (PART NO. 12943-16001)	INDEX (PART NO. 12943-16002)			
0	0	LBT	LDA CAX STX	LDA CAY STY	LDB CBX STX	LDB CBY STY
1	1	SBT	LDX CXA STA	LDY CYA STA	LDX CXB STB	LDY CYB STB
2	2	CBT	LDX XAX STX	LDY XAY STY	LDX XBX STX	LDY XBY STY
3	3	SFB	LDX ISX STX	LDY ISY STY	LDX DSX STX	LDY DSY STY
4	4	MBT	LDX ADX +N STX	LDY ADY +N STY	LDX ADX −N STX	LDY ADY −N STY
5	5	CMW	LDX LAX STA	LDY LAY STA	LDX LBX STB	LDY LBY STB
6	6	MVW	LDA LDX SAX	LDA LDY SAY	LDB LDX SBX	LDB LDY SBY
7	7	TBS	JLY STY	JLY I(2) STY	LDY JPY	
8	8	SBS	LDX A	LDX B	STX A	STX B
9	9	CBS	LAX M,I(1)	LAY M,I(2)	LDX M,I(3)	LDX A,I(1)
10	10	None	SAX MP	SAY MP	STX MP	STY MP
11	11	None	JLY MP	JPY MP		

Note: 1. If all bits of A-register are cleared, all tests will be executed.

2. N = 52525
M = Memory
I = Indirect
MP = Memory Protect

DIAGNOSTIC PERFORMANCE

SECTION IV

This section contains detailed descriptions of all tests, HALT codes, error and information messages.

4-1. TEST DESCRIPTION (INDEX)

4-2. TEST 0

This test checks the following instruction sequences:

LDA	LDA	LDB	LDB
CAX	CAY	CBX	CBY
STX	STY	STX	STY

The A- or B-register is loaded with an operand, copied to the X- or Y-register, and the X- or Y-register is stored to memory. The A/B-registers and X/Y-registers are checked for the correct contents, and if correct, another operand is applied. A total of 34 operand cases are applied to each sequence. The operands are shown below:

Test Operands

OCT 0	OCT 177776
OCT 1	OCT 177775
OCT 2	OCT 177773
OCT 4	OCT 177767
OCT 10	OCT 177757
OCT 20	OCT 177737
OCT 40	OCT 177677
OCT 100	OCT 177577
OCT 200	OCT 177377
OCT 400	OCT 176777
OCT 1000	OCT 175777
OCT 2000	OCT 173777
OCT 4000	OCT 167777
OCT 10000	OCT 157777
OCT 20000	OCT 137777
OCT 40000	OCT 77777
OCT 100000	OCT 177777

In addition to the testing mentioned above, the instruction sequence tested is checked to determine if the unused registers remain unchanged during the sequence execution. To do this, the first sequence is executed with B set to zero for the 34 operands followed by B set to all ones for the 34 operands. In the same manner, the second sequence is executed for B=0's and B=1's. The third and fourth sequences are tested for A=0's and A=1's. A total of 8 iterations on 34 operands are performed. If any errors occur, the E030 message is printed.

4-3. TEST 1

This test checks the following instruction sequences:

LDX	LDY	LDX	LDY
CXA	CYA	CXB	CYB
STA	STA	STB	STB

The X- or Y-register is loaded with an operand, copied to A or B, and A or B stored to memory. See test 0 for testing of unused registers.

4-4. TEST 2

This checks the following instruction instruction sequences:

LDX	LDY	LDX	LDY
XAX	XAY	XBX	XBY
STX	STY	STX	STY

The X- or Y-register is loaded with an operand, exchanged with the A- or B-register; then X or Y and A or B registers are stored and the results checked. All unused registers during a particular sequence are loaded prior to execution and checked after execution. See test 0 for testing of unused registers.

4-5. TEST 3

This test checks the following instruction sequences:

LDX	LDY	LDX	LDY
ISX	ISY	DSX	DSY
STX	STY	STX	STY

The X- or Y-register is loaded with an operand, then incremented or decremented, and the results stored and checked. For the operand case of -1_8 the ISX instruction is checked that it skips and the DSX instruction is checked that it skips for the case of $+1_{10}$. Error message E032 occurs for failure of these two cases. Error message E033 can occur for all other cases where a skip should not occur but did. All unused registers are checked prior to and after execution for correct contents. See test 0 for testing of unused registers.

4-6. TEST 4

This test checks the following instruction sequences:

LDX	LDY	LDX	LDY
ADX + N	ADY + N	ADX − N	ADY − N
STX	STY	STX	STY

N = 52525

The X- or Y-register is loaded with an operand, a positive or negative Number N is added to the X- or Y-register and the register stored in memory. The overflow and extend registers are saved prior to

execution and checked after execution for the correct values. Error message E031 is printed if an error occurs followed by the E030 message; otherwise, message E030 is printed if the error is an add result error. All unused registers are checked prior to and after execution for correct contents. See test 0 for testing of unused registers.

4-7. TEST 5

This test checks the following instruction sequences:

LDX	LDY	LDX	LDY
LAX	LAY	LBX	LBY
STA	STA	STB	STB

The X- or Y-register is loaded with an initial displacement of zero, the A- or B-register loaded from a memory location starting the operand table indexed by X or Y and the results stored to memory. X and Y are looped through a count of 0 to 33_{10} to cover all operand cases. All unused registers are checked prior to and after execution for correct contents. See test 0 for testing of unused registers.

4-8. TEST 6

This test checks the following instruction sequences:

LDA	LDA	LDB	LDB
LDX	LDY	LDX	LDY
SAX	SAY	SBX	SBY

The A- or B-register is loaded with an operand, the X or Y register is loaded with an initial displacement of zero, the A- or B-register stored to a memory location starting a buffer table of results indexed by X or Y. X and Y are looped through a count of 0 to 33_{10} to cover all operand cases. The buffer table is initially cleared before each test execution. Error message E030 occurs for any errors detected. The contents of memory (M) are also displayed as shown below. All unused registers are checked prior to and after execution for correct contents. See test 0 for testing of unused registers.

```
E030  LDA-LDX-SAX-ERROR
REG-ACT-EXP

A     000001    000001      B     000000    000000
X     000001    000001      Y     000000    000000
M     000000    000001
```

4-9. TEST 7

This test checks the following instruction sequences:

JLY	JLY I(2)	LDY
STY	STY	JPY

I = indirect

The jump and load Y instruction is checked for direct and two levels of indirect memory reference. The instruction is executed and the Y-register stored. The Y-register should be equal to the JLY instruction +1. If not error messages E042 and E044 will result. If the instruction fails to jump, error messages E041 and 43 will occur.

The JPY instruction is checked by loading Y with a destination address and executing the JPY 0 instruction. If the instruction fails to jump, error message E045 occurs. Y is checked to see that it has not changed as a result of JPY instruction. Unused registers are not checked in this test.

4-10. TEST 8

This test checks the following instruction sequences:

LDX A LDX B STX A STX B

Although these instructions are the same functionally as the CAX, CBX, CXA, CXB instructions, a different macro call is used. The A- or B-register is loaded and then copied to X in the first two cases. In the second two cases, X is loaded and copied to the A- or B-registers. Error messages E034 through 37 may occur if an error is detected. Only one operand case is used for each sequence and unused registers are not checked.

4-11. TEST 9

This test checks the following instruction sequences:

LAX M,I(1) LAY M,I(2) LDX M,I(3) LDX A,I(1)

I = indirect
M = memory

The LAX, LAY, LDX, LDX A instructions are checked for various levels of indirect addressing. Error message E040 may occur if an error is detected. Only one operand case is used for each sequence and unused registers are not checked.

4-12. TEST 10

This test checks the following instruction sequences:

SAX MP SAY MP STX MP STY MP

MP = memory protected

Memory protect is turned on and the SAX, SAY, STX, STY instructions checked that they do not violate memory and that an interrupt occurs. If an interrupt does not occur for the instruction, a memory protect interrupt is generated with a CLF 0 instruction and error message E050 is reported. If the interrupt occurs correctly with the violation register containing the correct address, the contents of memory are checked to ensure that it has not changed. Error message E051 occurs if memory is changed. Only one operand case is used for each sequence and unused registers are not checked. If during this test, bit 15 of the violation register is set, a HALT with MDR = 106005_8 will occur with B containing the violation register. Continued program execution is inhibited by a jump to the error halt. A-register contains test/subtest number.

4-13. TEST 11

This test checks the following instruction sequences:

JLY MP JPY MP

MP = memory protected

Memory protect is turned on, Y loaded with zero and a JLY instruction executed to check that a memory protect interrupt occurs. If the JLY instruction fails to cause an interrupt (the instruction jumps) a memory protect interrupt is forced with a CLF 0 instruction and error message E054 results. The contents of the Y register is not checked. Only one operand case is used for each sequence and unused registers are not checked. The JPY instruction is checked in a similar manner. Error message E056 results if an error occurs in the JPY execution. If during this test, bit 15 of the violation register is set, a HALT with MDR = 106005_8 will occur with B containing the violation register. Continued program execution is inhibited by a jump to the error halt.

4-14. ERROR INFORMATION MESSAGES/HALT CODES (INDEX)

Table 4-1 is a list of HALT codes, program/test section, messages, and explanations of the messages.

4-15. TEST DESCRIPTION (WORD-BYTE-BIT)

Refer to Appendix A for the values and patterns used for each test.

The ability of memory protect to protect memory is tested by CBS, MBT, MVW, SBS, and SBT instructions. Error message E012 results if an error is detected.

The interrupt characteristics of the CBT, CMW, MBT, MVW, and SFB are tested. Error messages E013 or E014 are reported if an error is detected.

4-16. TEST 0, LBT

This test checks the operation of the load byte instruction (LBT). Nine operand cases are used for testing. The A-Register is checked for the correct byte to be loaded and the B-Register is checked that the byte address is incremented after A is loaded. Error message E005 or E006 results if an error is detected.

4-17. TEST 1, SBT

This test checks the operation of the store byte instruction (SBT). Eight operand cases are used for testing. Memory is checked for the correct byte. Error E007 results if an error is detected. If the instruction is executed while memory protect is enabled, error E012 will result.

Table 4-1. Error, Information Messages and Halt Codes (Index)

HALT CODE	PROGRAM/ TEST SECTION	MESSAGE	COMMENTS
102075	Test Control	None	Test selection request resulting from switch bit 9 being set. Enter to A/B-registers the desired group of tests to be executed and press RUN.
102076	Test Control	None	End of test halt resulting from switch register bit 15 being set (A-register = test number). To continue press RUN.
102077	Test Control	PASS XXXXXX	Diagnostic run complete. (A-register = XXXXXX). Switch register options may be changed or test selection changed by setting bit 9 of switch register. To continue press RUN.
106077	Test Control	None	HALT stored in location 2_8-77_8 to trap interrupts which may occur unexpectedly because of hardware malfunctions. M-register contains the I/O slot which interrupted. Diagnostic may be partially destroyed if HALT occurs. The program may have to be reloaded; the problem should be corrected before proceeding.
None	Test Control	EIG (INDEX) DIAGNOSTIC	Introductory message.
None	Test Control	TEST XX	Information message before error message (XX = test number). Message occurs for the first error within a test but is suppressed for any subsequent messages within the same test.
102030	Test 0-6	E030 XXX-YYY-ZZZ-ERROR REG-ACT-EXP A AAAAAA AAAAAA B BBBBBB X XXXXXX XXXXXX Y YYYYYY YYYYYY M MMMMMM MMMMMM (standard in test 6)	Error resulting from the failure of instruction sequence XXX, YYY, ZZZ to execute correctly. XXX, YYY, ZZZ given in table 2-1, test 0-6. All unused registers are checked in tests 2-6 but only the unused A, B-registers are checked in test 0,1.
102031	Test 4	E031 OVERFLOW-EXTEND ERROR REG-ACT-EXP OV XX EX YY	Error in overflow or extend bit. Push RUN to get E030 error report.
102032	Test 3	E032 {ISX, ISY, DSX, DSY} INSTR FAILED TO SKIP	Error resulting from increment/decrement and skip index instruction not skipping when it should.
102033	Test 3	E033 {ISX, ISY, DSX, DSY} INSTR SKIP'D BUT SHOULD NOT	Error resulting from increment/decrement and skip index instruction skipping when it should not.
102034	Test 8	E034 LDX A FAILED	LDX A failed to execute correctly. See test description.
102035	Test 8	E035 LDX B FAILED	LDX B failed to execute correctly. See test description.
102036	Test 8	E036 STX A FAILED	STX A failed to execute correctly. See test description.
102037	Test 8	E037 STX B FAILED	STX B failed to execute correctly. See test description.
102040	Test 9	E040 {LAX, LAY, LDX} INDIRECT FAILED	Instruction failed to execute correctly for indirect addressing.
102041	Test 7	E041 JLY INSTR FAILED TO JMP	JLY instruction failed to jump. See test description.

Table 4-1. Error, Information Messages and Halt Codes (Index) (Continued)

HALT CODE	PROGRAM/ TEST SECTION	MESSAGE	COMMENTS
102042	Test 7	E041 JLY INSTR JMP'D BUT Y INCORRECT	JLY instruction jumped but Y was not loaded with the correct return address.
102043	Test 7	E043 JLY INSTR FAILED TO JMP,I	JLY instruction failed to jump indirect 2 levels. See test description.
102044	Test 7	E044 JLY INSTR JMP'D INDIRECT BUT Y INCORRECT	JLY instruction jumped indirect but Y was not loaded with the correct return address.
102045	Test 7	E045 JPY INSTR FAILED TO JMP	JPY instruction failed to jump. See test description.
102046	Test 7	E046 JPY INSTR JMP'D BUT Y INCORRECT	JPY instruction jumped but Y changed as a result of the jump.
102050	Test 10	E050 { SAX / SAY / STX / STY } INSTR DID NOT CAUSE MP INT	Store index instruction failed to cause memory protect interrupt.
102051	Test 10	E051 { SAX / SAY / STX / STY } INSTR CAUSED MP INT BUT MEMORY WAS NOT PROTECTED	Store index instruction caused a memory protect interrupt but failed to protect memory.
102054	Test 11	E054 JLY INSTR FAILED TO CAUSE MP INT	JLY instruction failed to cause memory protect interrupt. See test description.
102056	Test 11	E056 JPY INSTR FAILED TO CAUSE MP INT	JPY instruction failed to cause memory protect interrupt. See test description.
—	Test 10, 11	H047 MEMORY PROTECT OPTION NOT PRESENT	Memory protect option not configured during set up of diagnostic configurator.
106005	Test 10, 11	None	Memory parity error detected (bit 15 of violation register set) during memory protect testing. B-register contains contents of violation register. Diagnostic must be reloaded. A-register contains test/subtest number.
106010	Test 10	None	Memory protect option not configured in setup of Diagnostic Configurator or CLF0 did not interrupt.

4-18. TEST 2, CBT

This test checks the compare byte instruction (CBT) for three cases:

a. Byte string 1 equal to byte string 2.
b. Byte string 1 less than byte string 2.
c. Byte string 1 greater than byte string 2.

Error message E003 results if the instruction does not skip properly.

4-19. TEST 3, SFB

This test checks the scan byte instruction (SFB) for two cases:

a. Test byte equal to string byte.
b. Termination byte equal to string byte.

Error messages E003 or E011 are reported if an error is detected.

4-20. TEST 4, MBT

This test checks the move byte instruction (MBT) for a contiguous string of bytes. Error message E010 results if the bytes are not moved correctly. Error message E004 results if the instruction writes past the word count value.

4-21. TEST 5, CMW

This test checks the compare word instructions (CMW) for three cases:

a. String 1 equal to string 2.
b. String 1 less than string 2.
c. String 1 greater than string 2.

Error message E003 results if an error is detected.

4-22. TEST 6, MVW

This test checks the move word instruction (MVW) for a contiguous string of words. Error message E010 results if the words are not moved correctly. Error message E004 results if the instruction writes past the word count value.

4-23. TEST 7, TBS

This test checks the test bit instruction (TBS) for two test cases:

a. Test bit equal to mask bit.
b. Test bit not equal to mask bit.

Error message E003 results if an error is detected.

4-24. TEST 8, SBS

This test checks the set bit instruction (SBS) for the setting and resetting of all 16 bits. Error message E007 results if an error is detected.

4-25. TEST 9, CBS

This test checks the clear bit instruction (CBS). Error message E007 results if an error is detected.

4-26. ERROR INFORMATION MESSAGES/HALT CODES (WORD-BYTE-BIT)

Table 4-2 is a list of HALT codes, program/test section, messages, and explanations of the messages.

Table 4-2. Error, Information Messages and Halt Codes (Word-Byte-Bit)

HALT CODE	PROGRAM TEST SECTION	MESSAGE	COMMENTS
102073	Configuration	None	I/O select code entered at configuration invalid. Must be greater than 7_8. Reenter a valid select code and press RUN.
102074	Configuration	None	Select code entered during configuration valid. Enter program option bits to switch register and press RUN.
102075	Test Control	None	Test selection request resulting from switch bit 9 being set. Enter to A-register the desired group of tests to be executed and press RUN.
102076	Test Control	None	End of test halt resulting from switch register bit 15 being set (A-register = test number). To continue press RUN.
102077	Test Control	PASS XXXXXX	Diagnostic run complete. (A-register = XXXXXX). Switch register options may be changed or test selection changed by setting bit 9 of switch register. To continue press RUN.
106077	Test Control	None	HALT stored in location 2_8-77_8 to trap interrupts which may occur unexpectedly because of hardware malfunctions. M-register contains the I/O slot which interrupted. Diagnostic may be partially destroyed if HALT occurs. The program may have to be reloaded; the problem should be corrected before proceeding.
None	Test Control	EIG (WORD.BYTE.BIT) DIAGNOSTIC DSN XXXXXX	Introductory message. DSN = XXXXXX is the Diagnostic Serial Number.
All Error Halts	0-9	IN TEST-CASE XXXXXX OF YYY INSTR	Message attached to all error messages. Value XXXXXX equals test case number and YYY is the instruction under test (see Appendix A).
None	1, 4, 6, 8, 9	H001 MEMORY PROTECT NOT PRESENT	Memory protect option not configured during set up of diagnostic configurator.

Table 4-2. Error, Information Messages and Halt Codes (Word-Byte-Bit) (Continued)

HALT CODE	PROGRAM TEST SECTION	MESSAGE	COMMENTS
102003	2, 3, 5, 7	E003 INSTR DOES NOT SKIP PROPERLY	The instruction tested failed to branch to the correct exit (i.e., P+1, P+2, or P+3).
102004	4, 6	E004 TEST INSTR EXECUTED PAST SPECIFIED WORD COUNT	The instruction tested exceeded the word/byte count value.
102005	0	E005 WRONG RESULT RETURNED IN A-REG	LBT instruction failed to return correct value in A-register.
102006	0	E006 WRONG RESULT RETURNED IN B-REG	LBT instruction failed to return correct byte address in B-register.
102007	1, 8, 9	E007 WRONG DATA RETURNED	The instruction tested failed to return the correct data value. The B-register contains the actual value returned and the A-register contains the expected result.
102010	4, 6	E010 DESTINATION STRING DOES NOT COMPARE WITH SOURCE STRING	The instruction tested failed to move data correctly to the destination point.
102011	3	E011 INCORRECT TERMINATION	The SFB instruction did not exit properly when test or termination byte encountered.
102012	1, 4, 6, 8, 9	E012 MEMORY PROTECT DIDN'T OCCUR	Instruction tested failed to cause memory protect interrupt.
102013	2, 3, 4, 5, 6	E013 INTP DIDN'T OCCUR	Instruction tested could not be interrupted.
102014	2, 3, 4, 5, 6	E014 INTP FROM WRONG LOCATION	As interrupt did not occur at the address of the instruction under test.

WORD-BYTE-BIT TEST CASES

APPENDIX A

NOTE

All values given are in octal unless otherwise specified.

A-1. TEST 0, LBT — LOAD BYTE

The byte address is contained in the B-register with bit 0 indicating left (0) or right (1) byte. Note that the LBT instruction increments the B-register. To determine the actual byte address for the test-case used, decrement the B-register.

TEST-CASE	BYTE DEFINED IN B-REG	CONTENTS ADDRESSED BY B-REG	EXPECTED RESULT IN A-REG
1	DBL	125377	252
2	DBR	125377	377
3	DBL	177652	377
4	DBR	177652	252
5	DBL	052777	125
6	DBR	052777	377
7	DBL	177525	377
10	DBR	177525	125
11	DBL	000000	0

A-2. TEST 1, SBT — STORE BYTE

The B-register contains the actual value returned.

TEST-CASE	A-REG	BYTE ADDR IN B-REG	EXPECTED RESULT IN A-REG
1	252	DBL	125000
2	252	DBR	125252
3	125	DBL	052652
4	125	DBR	052525
5	252	DBL	125125
6	0	DBR	125000
7	0	DBL	000000
10*	252	DBL	000000

*Instruction tested with memory protect.

A-3. TEST 2, CBT — COMPARE BYTE

The byte address is contained in the A- and B-registers. Bit 0 equal to 0 defines a left byte address (DBL); equal to 1 defines a right byte address (DBR). Note that the CBT instruction increments the A- and B-registers. To determine the actual byte addresses for the test-case used, decrement the A- and B-registers.

TEST-CASE	ASCII STRING ADDRESSED BY		BYTE ADDR IN A/B-REG	SKIP RESULT
	A-REG (S1)	B-REG (S2)		
1	12345670	12345670	DBL	S1 = S2
2	12345670	12347670	DBR	S1 < S2
3	12345670	12345570	DBR	S1 > S2

A-4. TEST 3, SFB — SCAN FOR BYTE

The A-register contains the termination byte (high-order byte) and the test byte (low-order byte). The B-register contains the first byte address of the string to be scanned.

TEST-CASE	TERMINATION BYTE	TEST BYTE	BYTE ADDR	ASCII STRING	SKIP RESULT
1	0	Q	DBR	ABCDEFGHIJKLMNOP QRSTUVWXYZ1234567 890	NO SKIP (P + 1)
2	0	/	DBL	ABCDEFGHIJKLMNOP QRSTUVWXYZ1234567 890	SKIP (P + 2)
4	Same as TC1 except interrupt characteristic tested.				
5	Same as TC2 except interrupt characteristic tested.				

A-5. TEST 4, MBT — MOVE BYTES

The A-register contains the byte address of the source string. The B-register contains the byte address of the destination string.

TEST-CASE	BYTE ADDR IN A/B-REG	OCTAL VALUE ADDRESSED BY							
		A-REG				B-REG (EXPECTED VALUE)			
1	DBL	040461	041062	041463	042067	040461	041062	041463	042067
2	DBR	052501	041103	042125		125101	041103	042252	
3*	DBL	040461	041062	041463	042067	125252	125252	125252	125252
4	Same as TC1 except interrupt characteristic tested.								

*Instruction tested with memory protect.

A-6. TEST 5, CMW — COMPARE WORDS

The A-register contains the first word address of array 1 and the B-register contains the first word address of array 2.

TEST-CASE	ARRAY 1 (A1)	ARRAY 2 (A2)	SKIP RESULT
1	12345670 (ASCII)	12345670 (ASCII)	A1 = A2
2	12345670 (ASCII)	12347670 (ASCII)	A1 < A2
3	12345670 (ASCII)	12345570 (ASCII)	A1 > A2
4	052777	000001	A1 > A2
5	000001	052777	A1 < A2
6	000001	177777	A1 > A2
7	177777	000001	A1 < A2
10	100000	052777	A1 < A2
11	052777	100000	A1 > A2
12	100000	125000	A1 < A2
13	125000	100000	A1 > A2
14	052525	052777	A1 < A2
15	052777	052525	A1 > A2
16	177777	177777	A1 = A2
17	100000	100000	A1 = A2
20	000001	000001	A1 = A2
21	077777	077777	A1 = A2
22	000000	000000	A1 = A2

A-7. TEST 6, MVW — MOVE WORDS

The A-register contains the first word address of the source string. The B-register contains the first word address of the destination string. The word addresses in the A- and B-registers are incremented as each word is being moved.

TEST-CASE	STRING ADDRESSED BY	
	A-REG	B-REG (EXPECTED VALUE)
1	055060 054461 054111 053463	055117 054461 054111 053463
2*	055117 054461 054111 053463	125252 125252 125252 125252
3	Same as TC1 except interrupt characteristic tested.	

*Instruction tested with memory protect.

A-8. TEST 7, TBS — TEST BIT

The A-register contains the address of the mask and the B-register contains the address of the word to be tested.

TEST-CASE	MASK	VALUE TESTED	SKIP RESULT
1	125252	052525	SKIP
2	052525	125252	SKIP
3	125252	125252	NO SKIP
4	052525	052525	NO SKIP

A-9. TEST 8, SBS — SET BITS

The B-register contains the actual value returned.

TEST-CASE*	INITIAL BIT SETTING	MASK	EXPECTED RESULT IN A-REG
1	000000	125252	125252
2	000000	052525	052525
3	177777	125252	177777
4	177777	052525	177777
5	125252	052525	177777
6	052525	125252	177777
7	177777	000000	177777
10	000000	177777	177777
11	000000	000000	000000
12	000000	125252	125252
13	000000	052525	052525
14	177777	052525	177777
15**	000000	125252	000000
16**	000000	052525	000000
17**	177777	125252	177777

*Test-cases 1 thru 11 use indirect addressing whereas test-cases 12 thru 14 use direct addressing.
**Instruction tested with memory protect.

A-10. TEST 9, CBS — CLEAR BITS

The B-register contains the actual value returned.

TEST-CASE*	INITIAL BIT SETTING	MASK	EXPECTED RESULT IN A-REG
1	177777	125252	052525
2	177777	052525	125252
3	177777	177777	000000
4	177777	000000	177777
5	000000	000000	000000
6	000000	177777	000000
7	177777	125252	052525
10	177777	052525	125252
11	177777	177777	000000
12	177777	000000	177777
13	000000	000000	000000
14	000000	177777	000000
15**	177777	125252	177777

*Test-cases 1 thru 6 use indirect addressing whereas test-cases 7 thru 14 use direct addressing.
**Instruction tested with memory protect.

INSTRUCTION CODE AND MNEMONIC ASSIGNMENTS

APPENDIX B

Instruction Mnemonic	Description	Instruction Code	Instruction Mnemonic	Description	Instruction Code
ADX	Add Memory to X	105746	LBT	Load byte	105763
ADY	Add Memory to Y	105756	LBX	Load B indexed by X	105742
CAX	Copy A to X	101741	LBY	Load B indexed by Y	105752
CAY	Copy A to Y	101751	LDX	Load X from memory	105745
CBS	Clear bits	105774	LDY	Load Y from memory	105755
CBT	Compare bytes	105766	MBT	Move bytes	105765
CBX	Copy B to X	105741	MVW	Move words	105777
CBY	Copy B to Y	105751	SAX	Store A indexed by X	101740
CMW	Compare words	105776	SAY	Store A indexed by Y	101750
CXA	Copy X to A	101744	SBS	Set bits	105773
CXB	Copy X to B	105744	SBT	Store byte	105764
CYA	Copy Y to A	101754	SBX	Store B indexed by X	105740
CYB	Copy Y to B	105754	SBY	Store B indexed by Y	105750
DSX	Decrement X and skip if zero	105761	SFB	Scan for byte	105767
DSY	Decrement Y and skip if zero	105771	STX	Store X to memory	105743
ISX	Increment X and skip if zero	105760	STY	Store Y to memory	105753
ISY	Increment Y and skip if zero	105770	TBS	Test bits	105775
JLY	Jump and Load Y	105762	XAX	Exchange A and X	101747
JPY	Jump Indexed by Y	105772	XAY	Exchange A and Y	101757
LAX	Load A indexed by X	101742	XBX	Exchange B and X	105747
LAY	Load A indexed by Y	101752	XBY	Exchange B and Y	105757

HEWLETT-PACKARD COMPANY
11000 WOLFE ROAD, CUPERTINO, CALIFORNIA, 95014

MANUAL PART NO. 12943-90004
MICROFICHE PART NO. 12943-90005

Printed: SEPT 1978
Printed in U.S.A.